

IN THE CLAIMS:

Please amend the claims as follows.

1. (Canceled)
2. (Currently Amended) The method of claim 5, further comprising generating a visual map of the IT system, the visual map including a depiction of at least one of the discovered components and the at least one dependency between two or more of the discovered components.
3. (Currently Amended) The method of claim 2, wherein the visual map includes tracked changes to at least one of the discovered components.
4. (Currently Amended) The method of claim 5, wherein at least one of the discovered components is an application.
5. (Currently Amended) A method for collecting information on components in an information technology (IT) system, comprising:
discovering the existence of at least one of the components in the IT system by:
receiving one or more real-time messages corresponding to one or more occurrences in the IT system, wherein the real-time messages include event information relating to one or more components, including a first component; and
comparing at least the first component to one or more fingerprints, wherein each fingerprint corresponds to a known component and includes one or more attributes that are a subset of attributes in a model of the known component; and ~~using at least one fingerprint and receiving the real time messages as event information regarding an occurrence in the IT system, the occurrence relating to a first component, comparing the first undiscovered component along with other components to the at~~

~~least one fingerprint, wherein the fingerprint represents key low level elements of a model of a known component, and~~

~~based on said comparing, matching one or more components with one or more fingerprints determining that at least one of the components exists when all of the elements of the fingerprint corresponding to the known component are matched;~~
determining at least one dependency between two or more of the discovered components; and
tracking changes to at least one of the discovered components and the dependency between two or more of the discovered components.

6. (Currently Amended) The method of claim 5, wherein the indicated occurrence is selected from one or more of a file creation, a file deletion, and a file modification.

7. (Currently Amended) The method of claim 5, wherein the indicated occurrence is selected from one or more of a registry key creation, a registry key deletion, and a registry key modification.

8. (Currently Amended) The method of claim 5, wherein the indicated occurrence is information regarding detection of a particular component in the IT system.

9. (Currently Amended) The method of claim 5, further comprising indicating that a particular component has been damaged if the indicated occurrence is a deletion and at least one of the ~~elements~~ attributes of the fingerprint ~~is~~ are no longer matched by the components in the IT system.

10. (Currently Amended) The method of claim 5, further comprising indicating that a particular component has been uninstalled if the indicated occurrence is a deletion and all of the attributes ~~elements~~ of a minimum set of the fingerprint are no longer matched by the components in the IT system.

11. (Previously Presented) The method of claim 5, wherein the at least one dependency is selected from the group consisting of shared library usage, network usage, and containment dependencies.

12. (Currently Amended) The method of claim 5, further comprising:
generating a component discovered message upon the discovery of one of the components;
retrieving a list of elements to track [[for]] the discovered component; and
using the list of elements to track changes to the discovered component.

13. (Currently Amended) A computer-readable medium storing program instructions that are computer executable to implement an ~~An~~ agent for collecting information on components in an information technology (IT) system, the agent residing on a computer in the IT system, wherein the program instructions are executable to implement the agent comprising:

- an observer module to detect real-time event information about elements of the computer;
- and
- an analysis module to process the real-time event information, the analysis module including: (a) component discovery rules to process real-time event information and match event information with elements of one or more fingerprints of known components using an accumulator to discover ~~the existence on the IT system of at least one of the~~ or more components, wherein each fingerprint corresponds to a known component and includes one or more attributes that are a subset of attributes in a model of the known component, and (b) dependency discovery rules to detect relationships between discovered components of the IT system.

14. (Currently Amended) A system for collecting information on components in an information technology (IT) system, comprising:

- means for discovering the existence of at least one of the components in the IT system by:
 - receiving one or more real-time messages corresponding to one or more occurrences in the IT system, wherein the real-time messages include event information ~~and using at least one fingerprint and by receiving the real-time messages as event information~~ regarding an occurrence in the IT system, the occurrence relating to a first component[.];
 - comparing at least the first component to one or more fingerprints, wherein each fingerprint corresponds to a known component and includes one or more attributes that are a subset of attributes in a model of the known component; and
 - ~~comparing at least the first component along with other components to the at least one fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component,~~

based on said comparing, matching one or more components with one or more fingerprints and determining that at least one of the components exists when all of the elements of the fingerprint corresponding to the known component are matched;
means for determining at least one dependency between two or more of the discovered components; and
means for tracking changes to at least one of the discovered components and the dependency between two or more of the discovered components.

15. (Currently Amended) An apparatus for collecting information on components in an information technology (IT) system, comprising:

a memory storing a program;

a processor in communication with the memory; in which the processor is directed by the program to:

discover the existence of at least one of the components in the IT system by:

receiving one or more real-time messages corresponding to one or more occurrences in the IT system, wherein the real-time messages include event information and using at least one fingerprint and by receiving the real-time messages as event information regarding an occurrence in the IT system, the occurrence relating to a first component[[,]];

comparing at least the first component to one or more fingerprints, wherein each fingerprint corresponds to a known component and includes one or more attributes that are a subset of attributes in a model of the known component; and comparing the first component along with other components to the at least one fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component;

based on said comparing, matching one or more components with one or more fingerprints and determining that at least one of the components exists when all of the elements of the fingerprint corresponding to the known component are matched;
determine at least one dependency between two or more of the discovered components;
and

track changes to at least one of the discovered components and the dependency between two or more of the discovered components.

16. (Currently Amended) A method for discovering components in an information technology (IT) system, comprising:

receiving event information regarding an occurrence in the IT system, the occurrence relating to a first component;

comparing at least the first component ~~along with other components~~ to at least one fingerprint, wherein the fingerprint includes one or more attributes that are a subset of attributes in ~~represents key low-level elements of~~ a model of a known component; and

if any of the first compared components ~~and the other discovered components~~ match substantially all of the ~~key low-level elements~~ attributes of the fingerprint, using a subfingerprint of a refinement of the known ~~refined~~ component to discover the existence of a second component that corresponds to the refinement of the known ~~refined~~ component.

17. (Currently Amended) The method of claim 16, wherein the refinement of the known ~~refined~~ component is a version of the known component.

18. (Currently Amended) The method of claim 16, wherein the refinement of the known ~~refined~~ component is an optional piece of the known component.

19. (Currently Amended) The method of claim 16, further comprising generating a command message to collect further information if all of the ~~low-level elements~~ attributes of the fingerprint are matched.

20. (Currently Amended) The method of claim 19, further comprising receiving event information in response to the command message, wherein the event information is used with the subfingerprint of the refinement of the known ~~refined~~ component to discover the existence of the second component.

21. (Original) The method of claim 16, further comprising detecting low-level items in the IT systems and generating event information regarding the low-level items.

22. (Original) The method of claim 21, wherein the low-level items are selected from one or more of files, registry settings, and database schemas.

23. (Currently Amended) A computer-readable medium for discovering components in an information technology (IT) system, the computer-readable medium storing instructions that direct a microprocessor to:

receive event information regarding an occurrence in the IT system, the occurrence relating to a first component;

compare at least the first component ~~along with other components~~ to at least one fingerprint, wherein the fingerprint includes one or more attributes that are a subset of attributes in ~~represents key low level elements of~~ a model of a known component; and

if any of the first compared components ~~and the other discovered components~~ match substantially all of the ~~key low level elements~~ attributes of the fingerprint, ~~us[[e]]lling~~ a subfingerprint of a refinement of the known ~~refined~~ component to discover the existence of a second component that corresponds to the refinement of the known ~~refined~~ component.

24. (Currently Amended) An apparatus for discovering components in an information technology (IT) system, comprising:
a memory storing a program;
a processor in communication with the memory; in which the processor is directed by the program to:

receive event information regarding an occurrence in the IT system, the occurrence relating to a first component;

compare at least the first component ~~along with other components~~ to at least one fingerprint, wherein the fingerprint includes one or more attributes that are a subset of attributes in ~~represents key low-level elements of~~ a model of a known component; and

if any of the first compared components ~~and the other discovered components~~ match substantially all of the ~~key low-level elements~~ attributes of the fingerprint, using a subfingerprint of a refinement of the known ~~refined~~ component to discover the existence of a second component that corresponds to the refinement of the known ~~refined~~ component.

25. (Currently Amended) A method for managing components in an information technology (IT) system, comprising:

receiving a first event message for a first occurrence in the IT system, the first occurrence relating to a first component;

if the first component matches at least one ~~low-level element~~ attribute of a fingerprint of a model of a known component, adding the first component to an accumulator[[]], wherein the fingerprint includes one or more attributes that are a subset of attributes in a model of a known component;

if all of the ~~low-level elements~~ attributes of the fingerprint have been matched by the first component and other components, generating a command to detect further information;

receiving, in response to the command, a second event message providing further details about one of the components; and

using a subfingerprint of a refinement of the known ~~refined~~ component and the further details about one of the discovered components to discover a refined component.

26. (Original) The method of claim 25, wherein the first occurrence is one of a file creation, file deletion, file modification, registry key creation, registry key modification, and registry key deletion.

27. (Original) The method of claim 25, further comprising:
generating a component detected message upon the discovery of the refined component;
retrieving a list of elements to track for the refined component; and
using the list of elements to track changes to the refined component.

28. (Original) A method for discovery of a refined component in an information technology (IT) system, comprising:
using a fingerprint of a model of a known component to discover an existing component in the IT system by matching passive elements in the fingerprint with event information of the IT system;
generating and transmitting a command message defined by active elements of the fingerprint to discover the refined component;
receiving event information relating to the active elements of the fingerprint of the known component; and
using a subfingerprint of the refined component to discover the refined component, the refined component relating to the known component, wherein the subfingerprint of the refined component becomes active upon the discovery of the existing component using the fingerprint.

29. (Original) The method of claim 28, wherein receiving event information relating to active elements includes receiving an event message.

30. (Currently Amended) A method for determining dependencies between ~~at least two~~ components in an information technology (IT) system, comprising:

- discovering ~~the~~ at least two components in the IT system;
- monitoring the usage of resources by the ~~two~~ discovered components in the IT system by receiving real-time messages; ~~and,~~
- if a resource is used by one or more of the ~~two~~ discovered components, generating a message indicating the use of that resource ~~by that component;~~
- accumulating each message indicating the use of one of the resources by one of the ~~two~~ discovered components;
- if the accumulated messages indicate that at least the two of the discovered components use the same resource, then indicating that a first dependency between these ~~two~~ components has been detected; and
- determining a direction of the first dependency between the ~~two~~ indicated components.

31. (Canceled)

32. (Currently Amended) The method of claim 30, wherein the indicated component is selected from the group consisting of an application, a network connection endpoint, and a server.

33. (Currently Amended) The method of claim 32, wherein at least one message indicates a network outbound connection by one of the two discovered components.

34. (Currently Amended) The method of claim 32, wherein at least one message indicates a network listener by one of the two discovered components.

35. (Currently Amended) The method of claim 32, wherein at least one message indicates a use of a file by one of the two discovered components.

36. (Currently Amended) The method of claim 30, further comprising tracking changes to the first dependency between the two discovered components.

37. (Currently Amended) The method of claim 30, wherein the first dependency is a containment dependency.

38. (Currently Amended) The method of claim 30, wherein the first dependency is a network dependency.

39. (Currently Amended) The method of claim 30, wherein the first dependency is a shared usage dependency.

40. (Currently Amended) An apparatus for determining dependencies between ~~at least two~~ components in an information technology (IT) system, comprising:

- a memory storing a program;

- a processor in communication with the memory; in which the processor is directed by the program to:

 - discover ~~the~~ at least two components in the IT system;

 - monitor the usage of resources by the ~~two~~ discovered components in the IT system by receiving real-time messages; ~~and,~~

 - if a resource is used by one or more of the ~~two~~ discovered components, generating a message indicating the use of that resource ~~by that component;~~

 - accumulate each message indicating the use of one of the resources by one of the two components;

 - if the accumulated messages indicate that at least ~~the~~ two of the discovered components use the same resource, then indicating that a first dependency between these ~~two~~ components has been detected; and

 - determine a direction of the dependency between the ~~two~~ indicated components.

41-44 (Canceled)